DRAWING NO.	REVISION	MODULE	LANGUAGE
BP-003	A	Circuit Logic	English

Logic Gate Arrays in Low-Cost Automated Circuits

Build a real decision-making circuit — one that senses darkness and responds automatically — using nothing but logic gate chips. No microcontroller, no code.

SKILL LEVEL: BEGINNER–INTERMEDIATE | BUILD TIME: 2 HOURS | EST. COST: $6–10

How this connects: BP-001 and BP-002 built automated behavior using a microcontroller running code. This blueprint builds the *same kind* of automated decision — "turn on a light when it's dark" — with zero code, using only logic gate hardware. You don't need to have built BP-001 or BP-002 first, but seeing both approaches is what makes this genuinely click.

01 Objective

Every microcontroller sketch you write is, underneath, a chain of logic decisions — `if this AND that, then do this`. In this blueprint you'll build those same decisions directly in hardware, using a single CMOS logic chip called the CD4011. You'll first prove out how AND, OR, NOT, and NAND gates behave using switches and an LED, then combine them into one practical circuit: an automatic light that turns on when it's dark *and* a manual override switch is enabled. By the end, you'll understand why engineers sometimes choose to skip the microcontroller entirely — and why almost everything you build later will be a hybrid of both approaches.

02 Core Concept: Logic Gates

A logic gate takes one or more digital inputs (each either HIGH/1 or LOW/0) and produces one digital output, following a fixed rule. The CD4011 chip contains four independent **NAND** gates — and NAND is special because you can build every other basic gate out of NAND gates alone, which is exactly what you'll do below.

AND
Output HIGH only if both inputs are HIGH

OR
Output HIGH if either input is HIGH

NOT
Output is the opposite of the input

NAND
Output LOW only if both inputs are HIGH

INPUT A	INPUT B	AND	OR	NAND
0	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

03 Bill of Materials

COMPONENT	QTY	APPROX. COST	NOTES
CD4011 Quad 2-Input NAND Gate IC	1	$0.50–1	CMOS logic chip, 14-pin DIP package
Breadboard	1	$2–3	Reuse from BP-001 if available
LDR (light-dependent resistor)	1	<$1	Acts as the darkness sensor
10kΩ resistor	1	<$1	Forms the LDR voltage divider
Slide or push-button switch	2	$1–2	One is the manual override; one is used for the truth-table demo
LED (any color)	1	<$1	
220Ω resistor	1	<$1	Protects the LED
9V battery + clip (or 5V USB power source)	1	$2–3	No microcontroller needed — any clean 5V–9V source works with the CD4011
Jumper wires	~12	$1–2	

04 Tools Required

- Breadboard and jumper wires only — no computer, no software, no soldering

SAFETY NOTES

Low-voltage build (5–9V), no shock risk. One CMOS-specific caution: chips like the CD4011 can be damaged by static electricity before they're wired in. Touch a grounded metal surface before handling the IC, and avoid working on thick carpet in dry weather. Always insert the IC with pin 1 (marked by a small notch or dot) oriented correctly — reversed power will destroy the chip instantly.

05 System Architecture

The CD4011 has 4 independent NAND gates sharing one power pin (14) and one ground pin (7). You'll use one NAND gate wired as a NOT gate (both inputs tied together) to invert the darkness signal, and combine it with a second NAND gate to build the final AND-like decision: *dark AND override-enabled → LED on*.

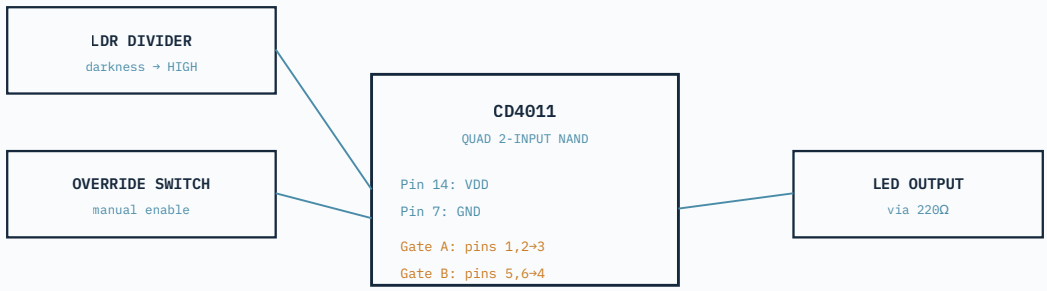

FIG. 1 — DARKNESS SENSOR + OVERRIDE SWITCH COMBINED THROUGH THE CD4011 TO DRIVE THE LED

06 Step-by-Step Build

1. **Place the CD4011 across the breadboard's center gap** so pins 1–7 are on one side and 8–14 on the other, straddling the divide.
2. **Power the chip.** Pin 14 → 5–9V positive rail. Pin 7 → ground rail.
3. **Part A — Prove the truth table.** Wire two switches to pins 1 and 2 (each switch connects its pin to either the positive rail or ground). Wire pin 3 (the output) through a 220Ω resistor to your LED, then to ground. Flip both switches through all four combinations and confirm the LED matches the NAND truth table in Section 02 — this step is really about seeing the logic behave before trusting it inside a bigger circuit.
4. **Wire the LDR darkness divider.** One leg of the LDR to the positive rail, the other leg to both a 10kΩ resistor (to ground) and to Gate B's input pin 5. Tie pin 5 and pin 6 together so Gate B acts as a single-input NOT gate — its output (pin 4) goes HIGH when it's dark.
5. **Wire the override switch** to Gate A's input pin 1. Connect Gate A's input pin 2 to Gate B's output (pin 4) — this is where the two signals combine.
6. **Wire Gate A's output (pin 3)** through the 220Ω resistor to the LED, then to ground.
7. **Test it.** Cover the LDR to simulate darkness, and flip the override switch. The LED should only light when both conditions are true — exactly like the AND row of the truth table.

07 Circuit Logic Walkthrough

There's no code to read here — the "program" is the wiring itself. Two NAND gates combine to behave like an AND gate: Gate B inverts the darkness signal into a clean digital HIGH/LOW, and Gate A combines that with the override switch. This two-NAND-gates-make-an-AND-gate trick is the same principle used to build every logic circuit that has ever existed, including the millions of gates inside the microcontroller you used in BP-001 and BP-002 — they're built from the exact same primitives, just packed a billion times smaller.

DARKNESS (GATE B OUT)	OVERRIDE SWITCH	LED (GATE A OUT)
LOW (bright)	OFF	OFF
LOW (bright)	ON	OFF
HIGH (dark)	OFF	OFF
HIGH (dark)	ON	**ON**

> Note: the LED brightness you get directly off a CD4011 output is dim — these chips can only source a few milliamps. That's fine for an indicator LED. To drive something bigger (a relay, a brighter light, a motor), you'd add a transistor between the gate output and the load — a natural next extension once this base circuit works.

08 Testing & Troubleshooting

SYMPTOM	LIKELY CAUSE	FIX
LED never lights, no combination works	IC damaged (often static discharge), or reversed power	Double-check pin 14/7 orientation before replacing the chip
LED stays on regardless of switches	An input pin is floating (not connected to HIGH or LOW)	Every gate input must be tied to something — never leave a CMOS input unconnected
Truth table demo (Part A) doesn't match Section 02	Output read from the wrong pin	Recheck the CD4011 pinout diagram; pin numbering starts at the notch
Darkness detection is unreliable / triggers randomly	LDR divider resistor value doesn't match your lighting conditions	Try a different resistor value (4.7kΩ–47kΩ) to shift the sensitivity threshold

09 Where This Goes Next

Why most real projects use both approaches

You've now built the same kind of automated decision two different ways: with code (BP-002) and with pure hardware logic (this blueprint). Hardware logic is faster and uses no power waiting on a processor, but it's fixed once wired — a microcontroller can change its behavior with new code alone. Most real engineering projects, including ISEF-level ones, use hardware logic for simple, fast, always-on decisions and a microcontroller for anything that needs flexibility or data logging. Knowing both — and being able to explain *why* you chose one over the other for a given part of your project — is exactly the kind of technical judgment that stands out in a research abstract. See the *ISEF & University Playbook* module for how to frame that design decision formally.